

# Control System Studio: BOY

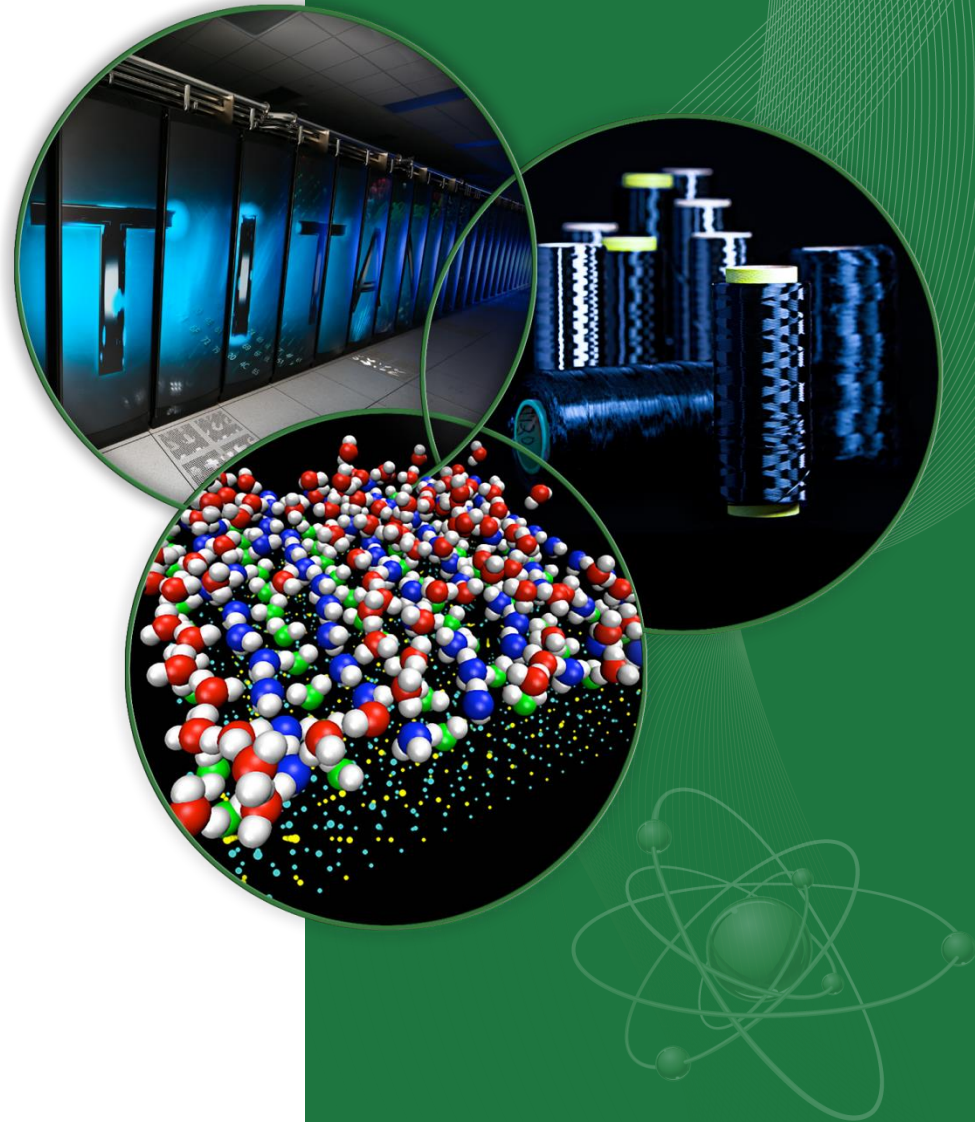
Kay Kasemir

ORNL/SNS

[kasemirk@ornl.gov](mailto:kasemirk@ornl.gov)

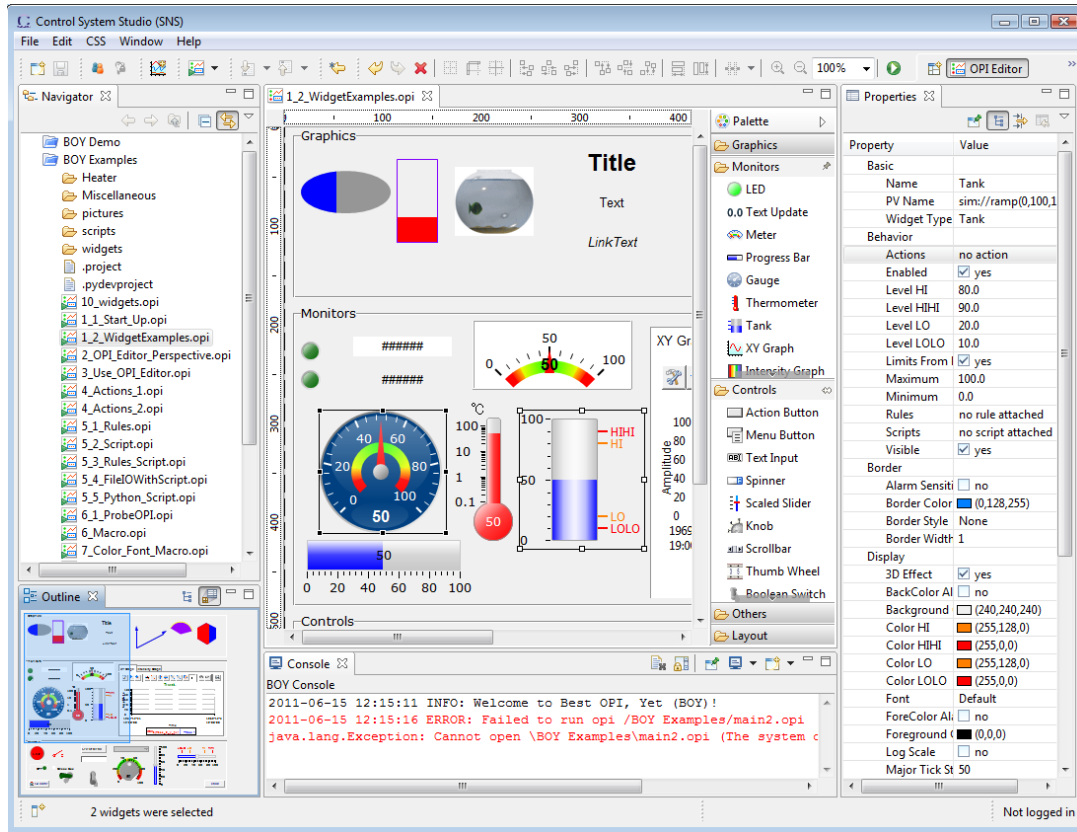
A lot of material from  
Nadine Utzel, ITER  
and BOY online help  
by Xihui Chen, SNS

June 2014

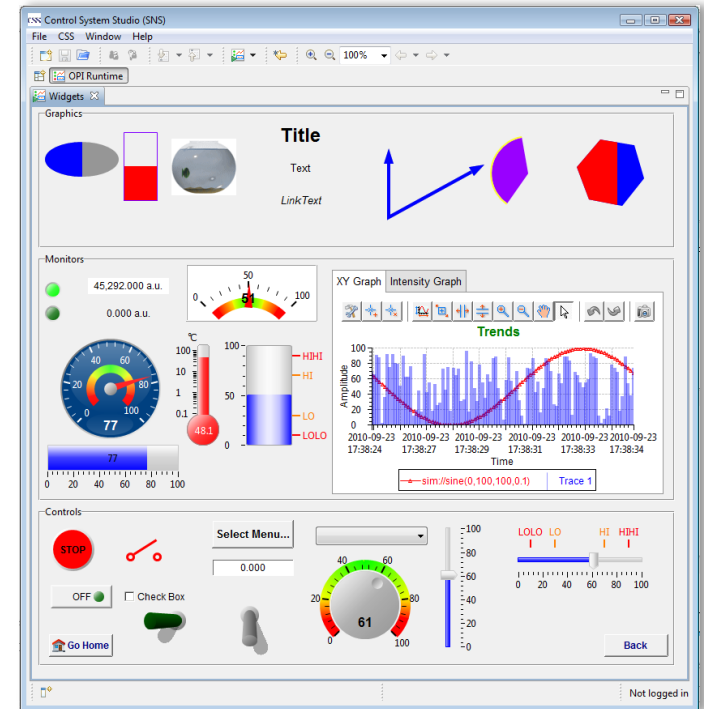


# BOY – Best OPI, Yet

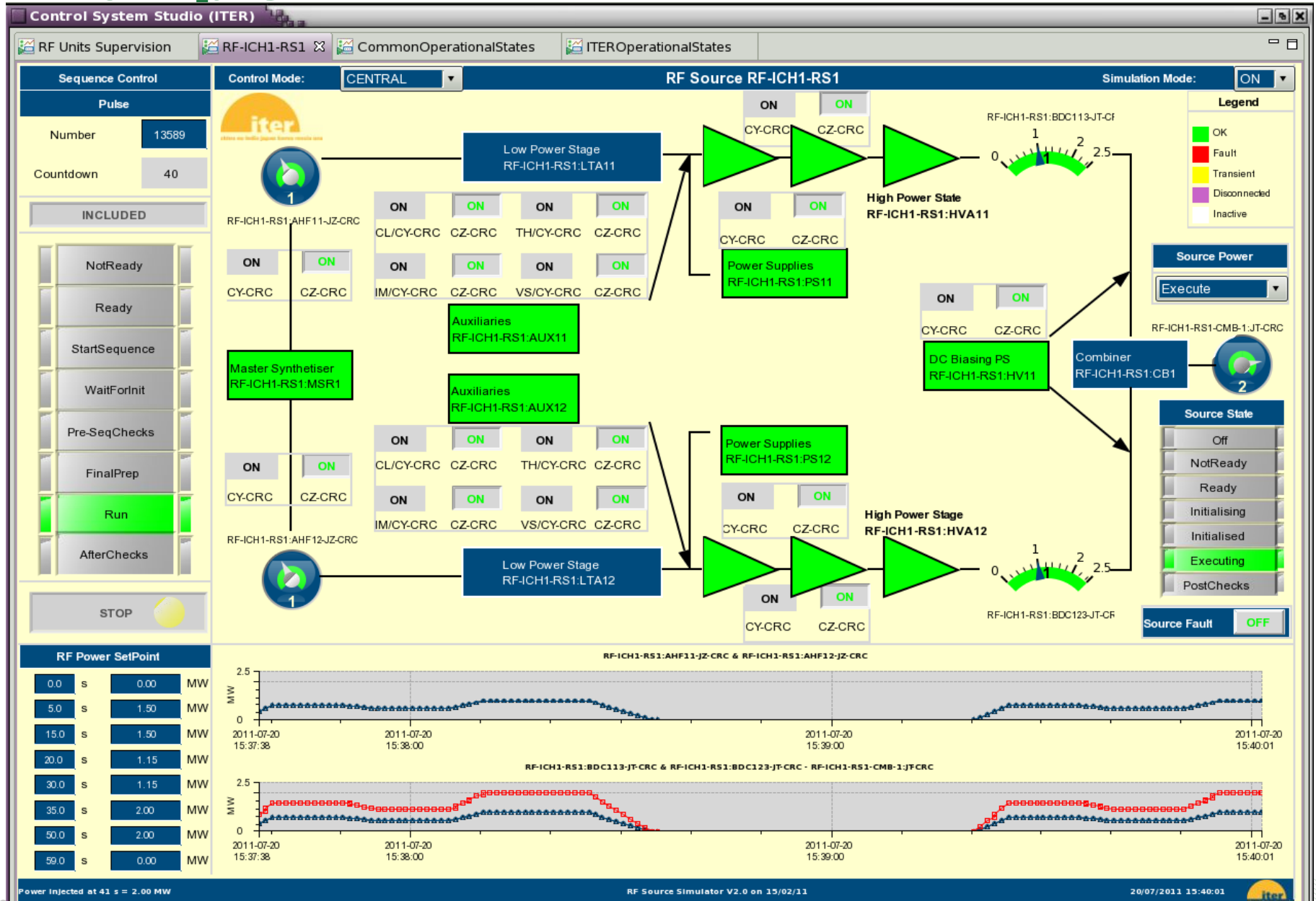
## Operator Interface Editor



## Runtime

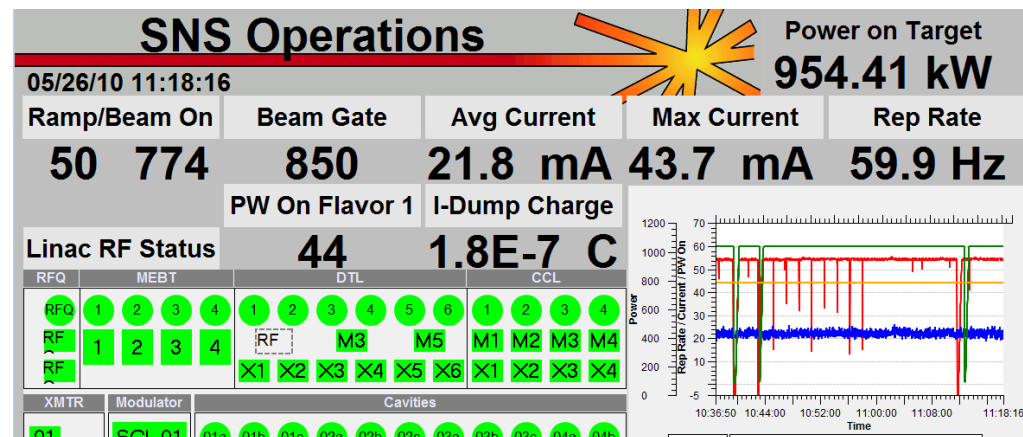


## Example: ITER



# Examples: SNS

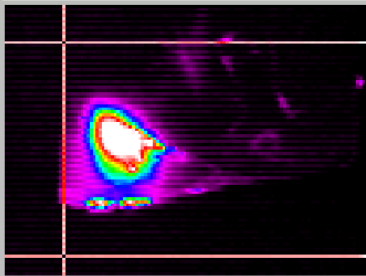
- Top-level displays



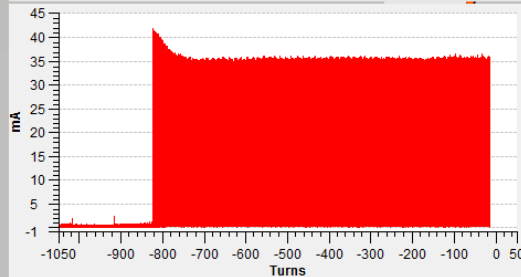
## SNS Central Control Room

05/26/10 10:37:56

Beam Image at Foil

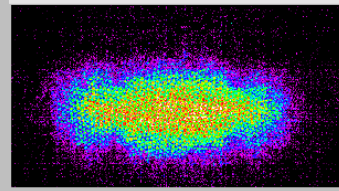


774 Bunches Energy 925 MeV



Power on Target **952 kW**

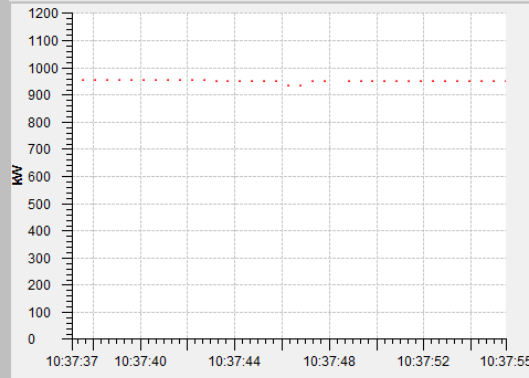
Rep Rate 59.9 Hz  
Beam To Target



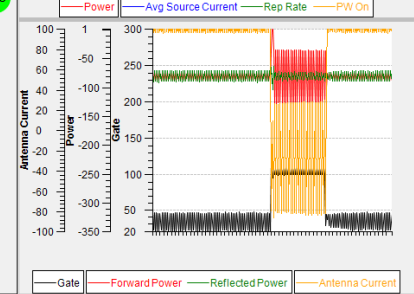
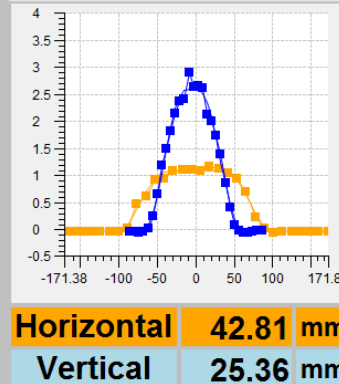
Primary Shutter Status

USANS	NOMAD	BASIS
SNAP	Magnetism	Liquids
CNCS	EQ-SANS	VULCAN
8	CORELLI	10
POWGEN	MaNDi	TOPAZ
FNPB	HYSPEC	NSE
VISION	SEQUOIA	ARCS

12-Hour Beam Power On Target



Beam Size at Target





# Examples: SNS

Control System Studio (SNS)

File Edit CSS Window Help

Test Bench ReadBack Test

## SNS Timing Receiver Test Bench


Card A (0x0) Card B (0x080000)

Board Info

Board ID	SNS Timing Receiver VME Board V2325	Board Rev	-	Base Address	0x0
Firmware Version	FW v 0.xxx Date 02 24 2011	Board SN	0x0	Geog Address	0x8

Event # 0-255	Delay Turns 0-65535	Delay 1/64th Turn 0-63	Delay Time us	Pulse Width 1/64th Turn 0-262143	Pulse Width Time us	Enable Output	Inverted Output	1 Shot Enable Output	Manual Fire
CH1	1	2	3	0000.000	4	0000.000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CH2	5	6	7	0000.000	8	0000.000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CH3	9	10	11	0000.000	12	0000.000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CH4	0	0	0	0000.000	0	0000.000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CH5	0	0	0	0000.000	0	0000.000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CH6	0	0	0	0000.000	0	0000.000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CH7	0	0	0	0000.000	0	0000.000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
CH8	3	0	0	0000.000	0	0000.000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ScratchPad

ScratchPad 0xC8	0xCFA71	0xCFA6D	Auto Test 
ScratchPad 0xCC	0xCFA71	0xCFA6D	
ScratchPad 0xD0	0xCFA71	0xCFA6D	
ScratchPad 0xD4	0xCFA71	0xCFA6D	

Grouping Container

Temp Limit Set 55.0

Temperature 25.38 C

Write Data to Arbitrary Address

Offset_Address	0xC8	0xC8	Data at Offset_Address:
Data to Write	0xCE263	0xCFA6D	31:24 23:16 15:8 7:0

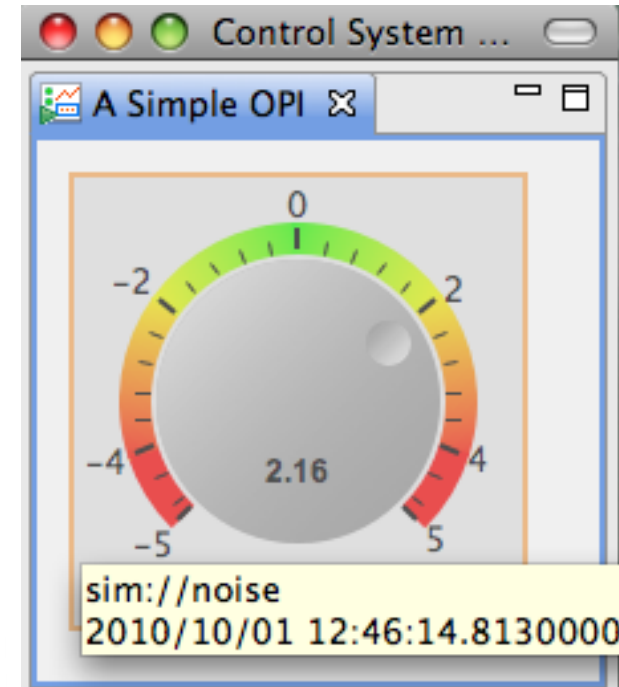
Not logged in

# Main Idea: Simple Things are Easy

1. Drag a widget, e.g. Knob, from palette to editor
2. Enter the PV name in Properties view
3. Click the “Run”  button to execute!

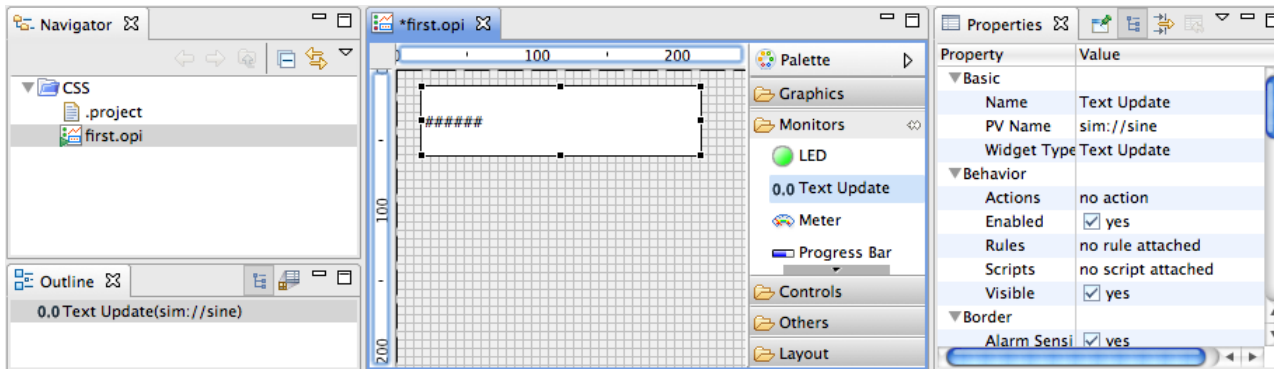
## What you will get


- ✓ *PV value* as text and via knob position
- ✓ *PV severity* reflected in border color
- ✓ *PV name* and *value* shown in tool-tip
- ✓ *PV display limits* set the knob's default range
- ✓ Indication of ‘disconnected’ state via a pink border
- ✓ Widget will be greyed-out if read-only

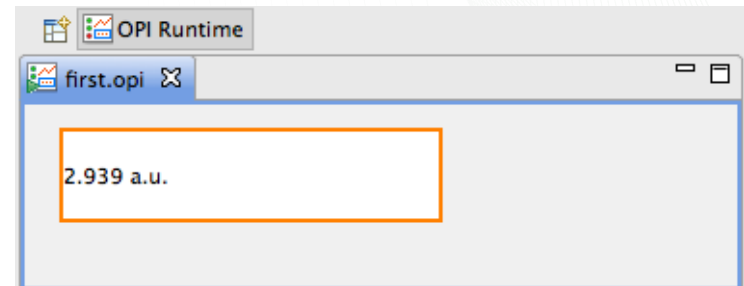


# First Display

- Menu CSS,
  - *Display, OPI Editor Perspective*
  - *Display, Install OPI Examples*
- Navigator Context menu on CSS: *New, OPI File*, call it “first.opi”
  - *Or Menu File, New, BOY, OPI File*




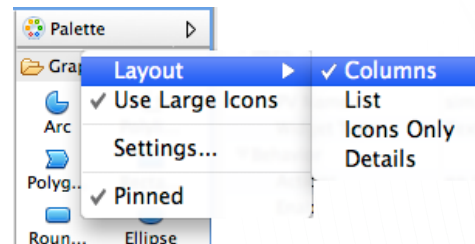
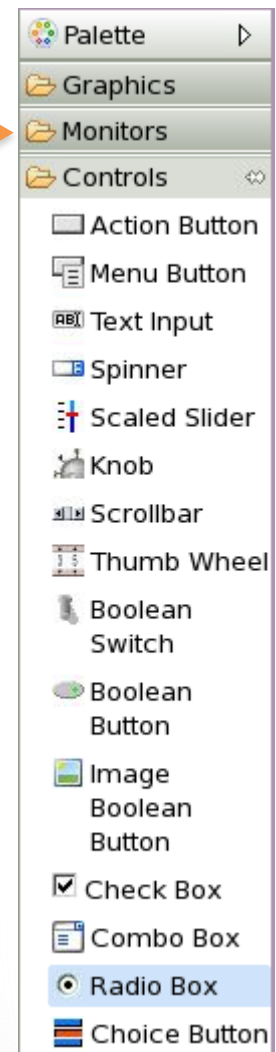
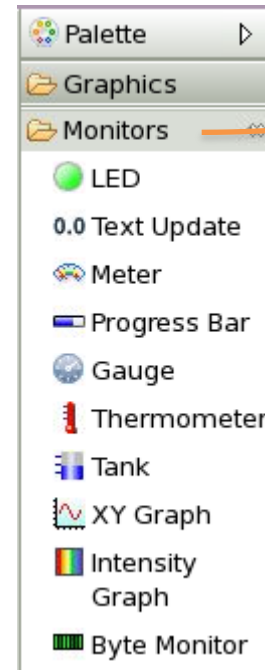
- Locate in Palette: *Monitors, Text Update*
  - ‘Drag’ *Text Update* onto display grid
  - Move widget around, resize
- Locate *Properties View*
  - Enter *PV Name* “sim://sine”
- Press Run  button in Toolbar



# Widget Palette Hints

Many widgets, hard to see them all

- Scroll
- Click on section header
- Try the 'pins' 
- Header Context menu offers *Columns* mode to display Widgets as small icons in columns

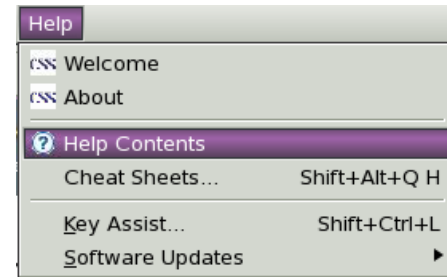




# View Online Help

## Find

- CSS Core, Process Variables
- CSS Applications, Display, BOY, Widgets



Help - CSS (ITER) (on next.codac.iter.org)

Search:  Go Scope: All topics

**Contents**

- CODAC Core System
- CSS Applications
  - Display
    - RDB Table Editor
    - adi2Boy - Converter
    - PV Table
    - Best OPI Yet
      - Introduction
      - Install Examples
      - Getting Started
      - OPI Editor Perspective
      - OPI Editor
      - OPI Runtime
    - Setting Preferences
    - Widgets
      - Widget Properties
      - PV Widgets
      - Shape Widgets
      - Label
      - Image
      - Boolean Widgets
      - Text Update
      - Scaled Widgets
      - XY Graph**
      - Intensity Graph
      - Byte Monitor
      - Action Button
      - Menu Button
      - Text Input
      - Spinner
      - Scrollbar
      - Thumb Wheel
      - Check Box
      - Choice Widgets
      - Container Widgets
      - Web Browser
      - Grid Layout
      - Color and Font
      - PV Connectivity
      - Action

CSS Applications > Display > Best OPI Yet > Widgets

### XY Graph

A widget that is able to plot 1D or 2D data in an XY Graph. It has comprehensive drawing and operating functionalities:

- Supports scalar PV, array or waveform PV.
- Line chart, scatter chart, bar chart, step chart, area chart...
- Abundant interactive operating capabilities: Five Zoom Types, Panning, Auto Scale, Add/Remove Annotations, Undo/Redo, Take snapshot.
- Configure properties at Runtime, such as changing trace color, line width and axis color etc.,
- Multiple axes support
- Log scale, date time format axis support
- Group legends by axes
- Annotations could be free or snapped to a trace
- ...

**Operations**

The widget is equipped with a toolbar which allows you to:

- Configure the properties of graph, axes or traces.
- Add/Remove Annotations. Annotations are moveable by dragging and dropping.
- Perform auto scaling.
- Zoom In/Out on plotting area or axes in different ways

# PV Names

- `ca://some_pv_name`
  - EPICS Channel Access PV
- `some_pv_name`
  - Typically same, since “ca://” is the default
- `sim://sine`
  - Simulated PV. Read online help for details
- `loc://x(4)`
  - Local PV. Read online help for details
- `pva://x`
  - EPICS V4 pvAccess

# Formula Support

= 'some\_pv\_name' \* 2

- Start with '='
- Enclose PV names in single quotes

= 3.14

- Formula with constant value  
(replaces previous 'const:\\3.14')

= "I like CS-Studio"

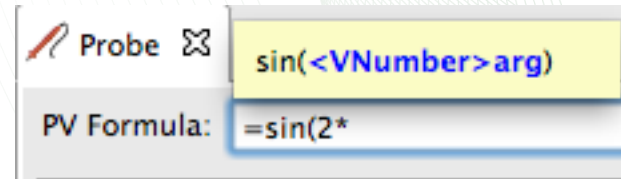
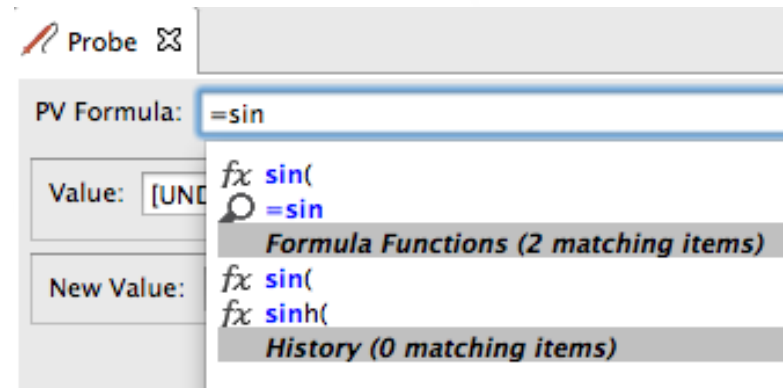
- Enclose strings in double quotes

loc: //x(4)

- Local PV. Read online help for details

Check online help,  
see CSS/Debugging/Formula,  
note auto-completion hints.

Beware:  
Don't use formulas for  
conversions that should  
happen on the IOC!



# Widget Properties


- Widgets are configured by setting Properties in the *Properties* view
- Common Properties:
  - Name
  - Position\*
  - Background color
  - Border
- Widgets that read/write PVs:
  - Basic: PV Name
  - Border: Alarm Sensitive
  - Behavior: Limits from PV

\* Position can also be modified by moving or resizing the widget in the editor, or via Toolbar buttons to align etc.

The screenshot shows a 'Properties' window for an 'LED' widget. The window has a title bar with a close button and a toolbar with icons for undo, redo, and other actions. Below the title bar is a tabbed interface with 'Property' and 'Value' tabs. The 'Property' tab is active, showing a list of properties grouped into sections: Basic, Behavior, Border, Display, and Position. Each property has a corresponding value, some of which are set to 'IO' (Input/Output) or 'Default'.

Property	Value
<b>Basic</b>	
Name	LED
PV Name	
Widget Type	LED
<b>Behavior</b>	
<b>Border</b>	
Alarm Sensitive	<input checked="" type="checkbox"/> yes
Border Color	<input type="checkbox"/> IO Border
Border Style	None
Border Width	1
<b>Display</b>	
3D Effect	<input checked="" type="checkbox"/> yes
BackColor Alarm Sensitive	<input type="checkbox"/> no
Background Color	<input type="checkbox"/> IO Background
Font	Default
ForeColor Alarm Sensitive	<input type="checkbox"/> no
Foreground Color	<input checked="" type="checkbox"/> IO Foreground
Off Color	(0,100,0)
Off Label	OFF
On Color	(0,255,0)
On Label	ON
Show Boolean Label	<input type="checkbox"/> no
Square LED	<input type="checkbox"/> no
Tooltip	\$(pv_name) / \$(p
<b>Position</b>	
Height	34
Width	34
X	167

# Extend First Display

- Locate in Palette: *Controls, Knob*
- Drag *Knob* onto display
- Move *Knob* around, resize
- Locate Property *PV Name* for Knob
- Enter “sim://sine”
- Create another *Knob*:
  - PV Name = “loc://test”,
  - “Increment” = 0.1
  - “Limits from PV” = no
- Run 
- Note how the “sim://sine” Knob is really read-only, but you can change the “loc://test” PV via the Knob



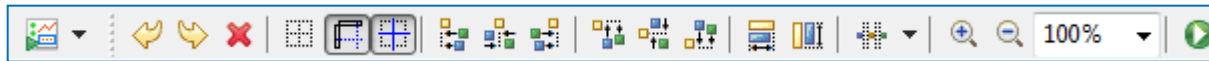


# Exercise: Editing Features

Add, duplicate Widgets in various ways

- Drag & Drop from Palette
- Copy/paste, Ctrl+Drag existing widgets to duplicate

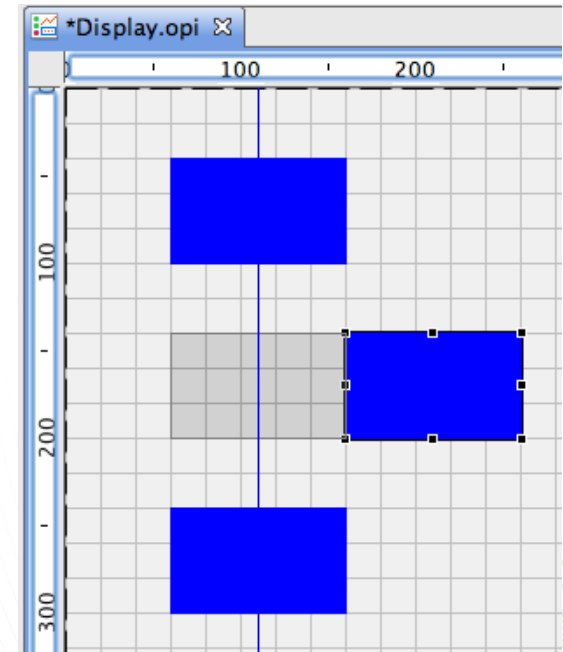
• Arrange them on the display



- Snap to grid, guideline, other widgets
- Align, distribute

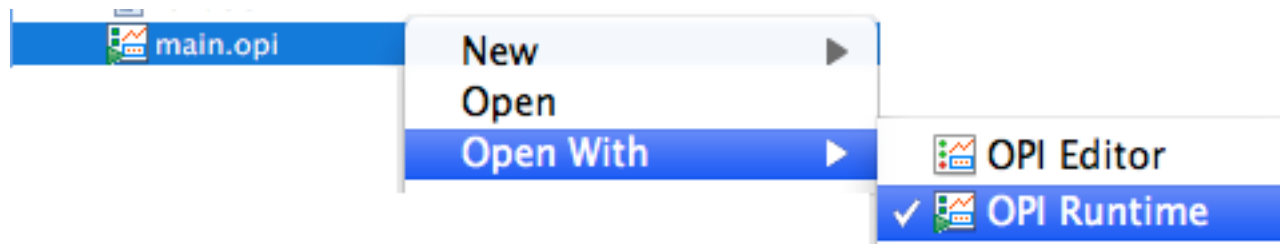
Select multiple widgets to

- Edit common properties
- Adjust size or move around



# OPI Files: Run or Edit?

- Default: Double-click on \*.opi in Navigator opens in “OPI Runtime”, i.e. executes the display
- Context menu allows to select
  - a) Editor to edit?
  - b) Runtime to execute?



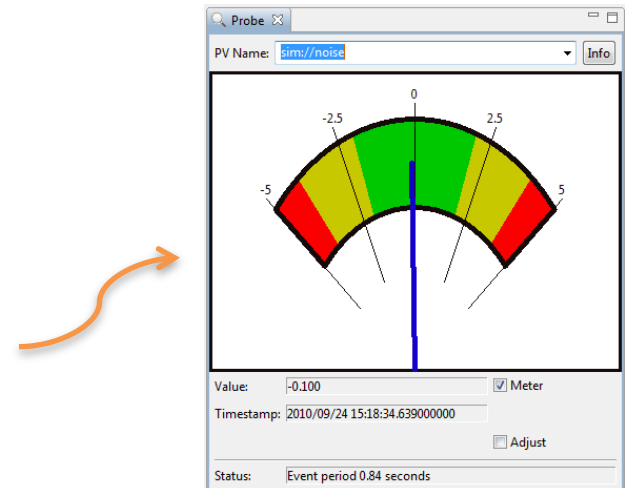
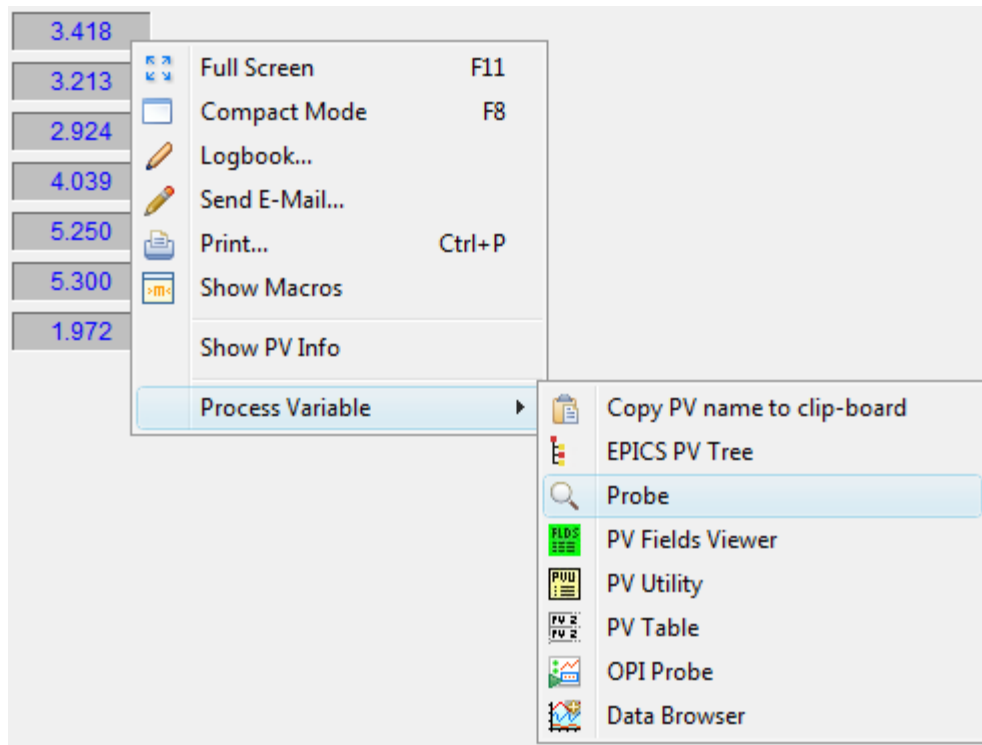
- Once you select “Editor”, that will become the double-click default
  - Select “Runtime” once to restore previous default

# Exercise: Edit vs. Runtime Mode

- Close all CSS Editors (Menu *File, Close All*)
- In the Navigator, double-click on the first.opi that you created before
  - Does it open in the Editor or Runtime?
- In the Navigator, open the Context Menu on first.opi and select Open With, OPI Editor.
  - Close first.opi, now double-click the file in the Navigator. Does it open in the Editor?
- In the Navigator, open the Context Menu on first.opi and select Open With, OPI Runtime.
  - Close first.opi, now double-click the file in the Navigator. Does it open in the Runtime?

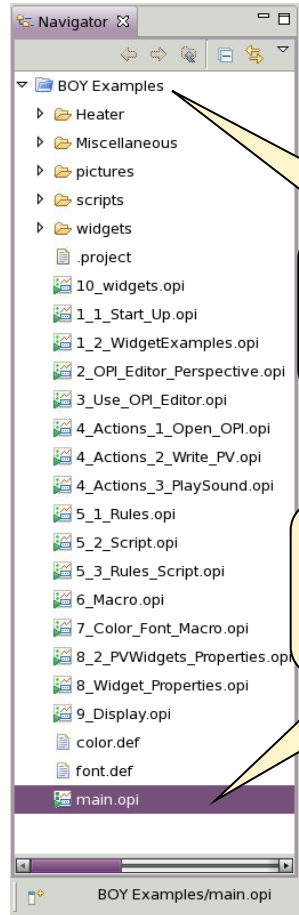
# Exercise: Send PV to other CSS tools

- Run the OPI that you created
- Use CSS Process Variable context menu on a widget that displays a PV to open Probe



# Example Displays

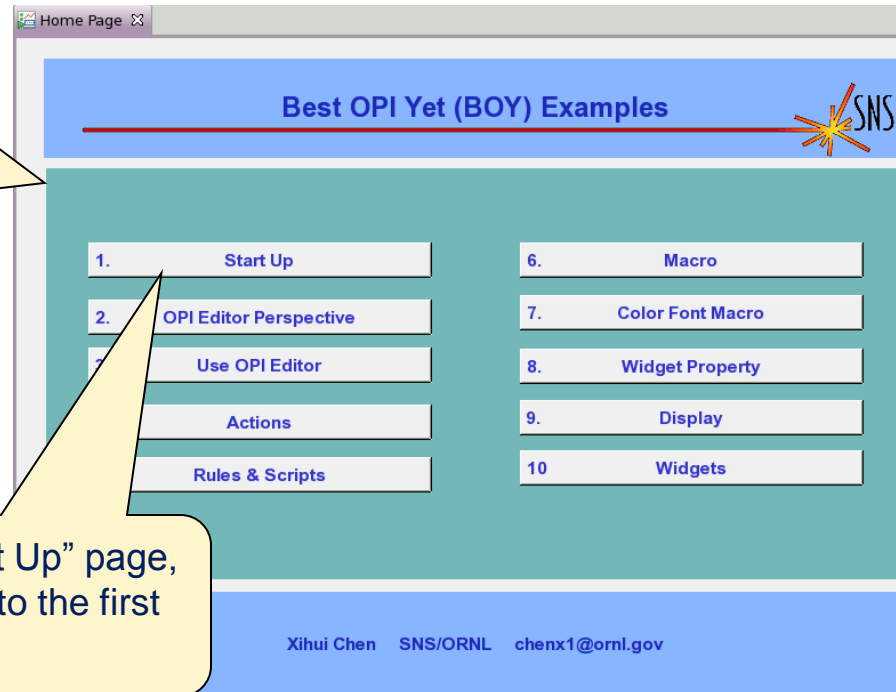
- Installed via Menu *CSS, Display, Install OPI Examples*



Note new project named BOY Examples

Explore the examples

Double-click on main.opi file to open



1. Start Up

2. OPI Editor Perspective

3. Use OPI Editor

4. Actions

5. Rules & Scripts

6. Macro

7. Color Font Macro

8. Widget Property

9. Display

10. Widgets

Xihui Chen SNS/ORNL chenx1@ornl.gov

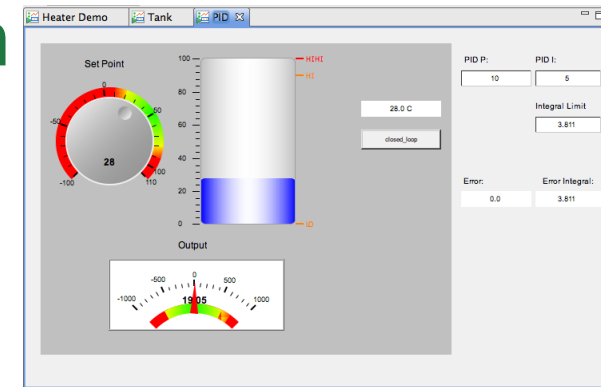
Check the "Start Up" page, which is similar to the first two exercises

Remember: You can *Open With, .. Editor* to see implementation



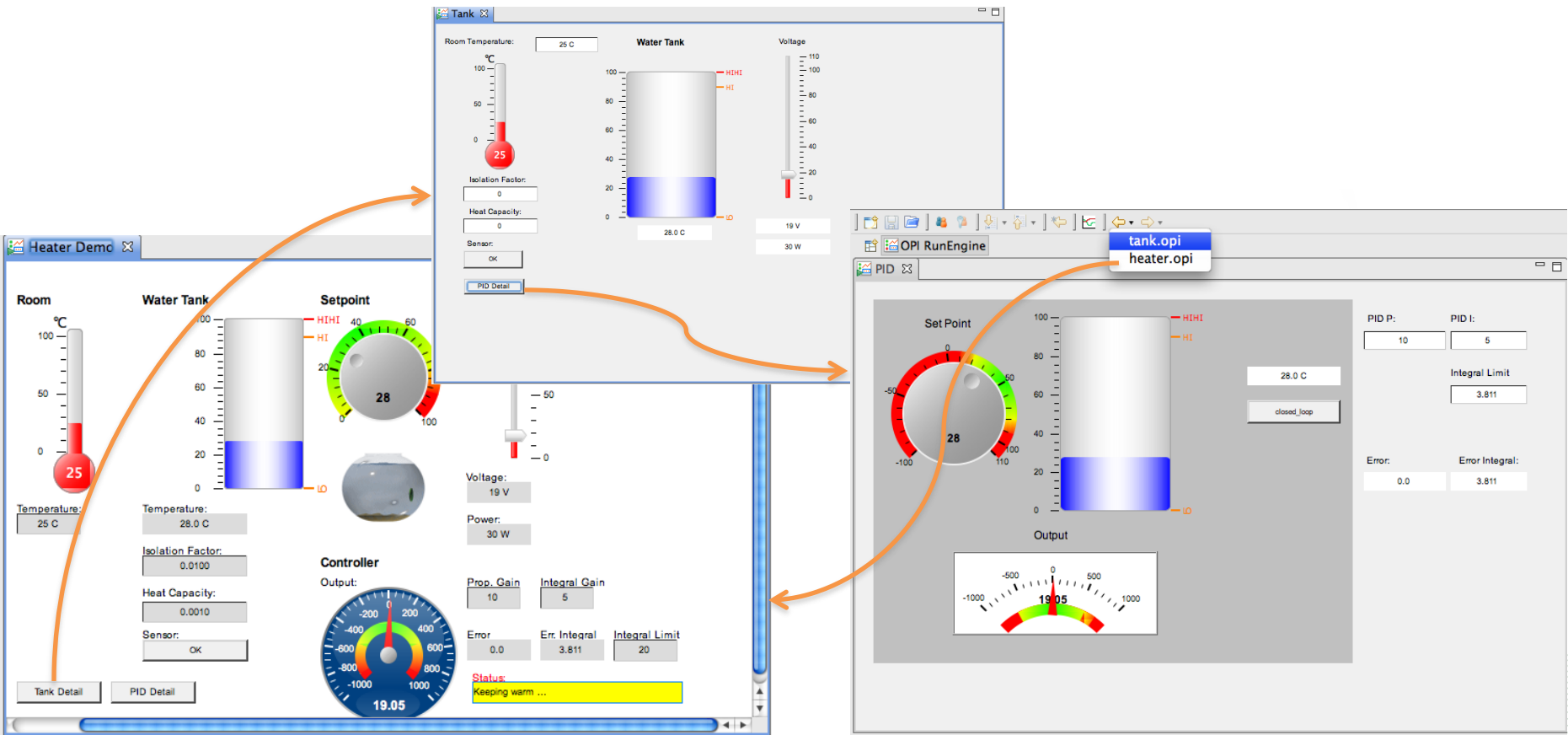
# Exercise: Screen Navigation

- Similar to hyperlinks in a [Web Browser](#):
  - Default: Linked display replaces the current display.
  - Zoom in/out, go “back” via toolbar:
  - Use context menu to open in ‘tabs’ or new Window



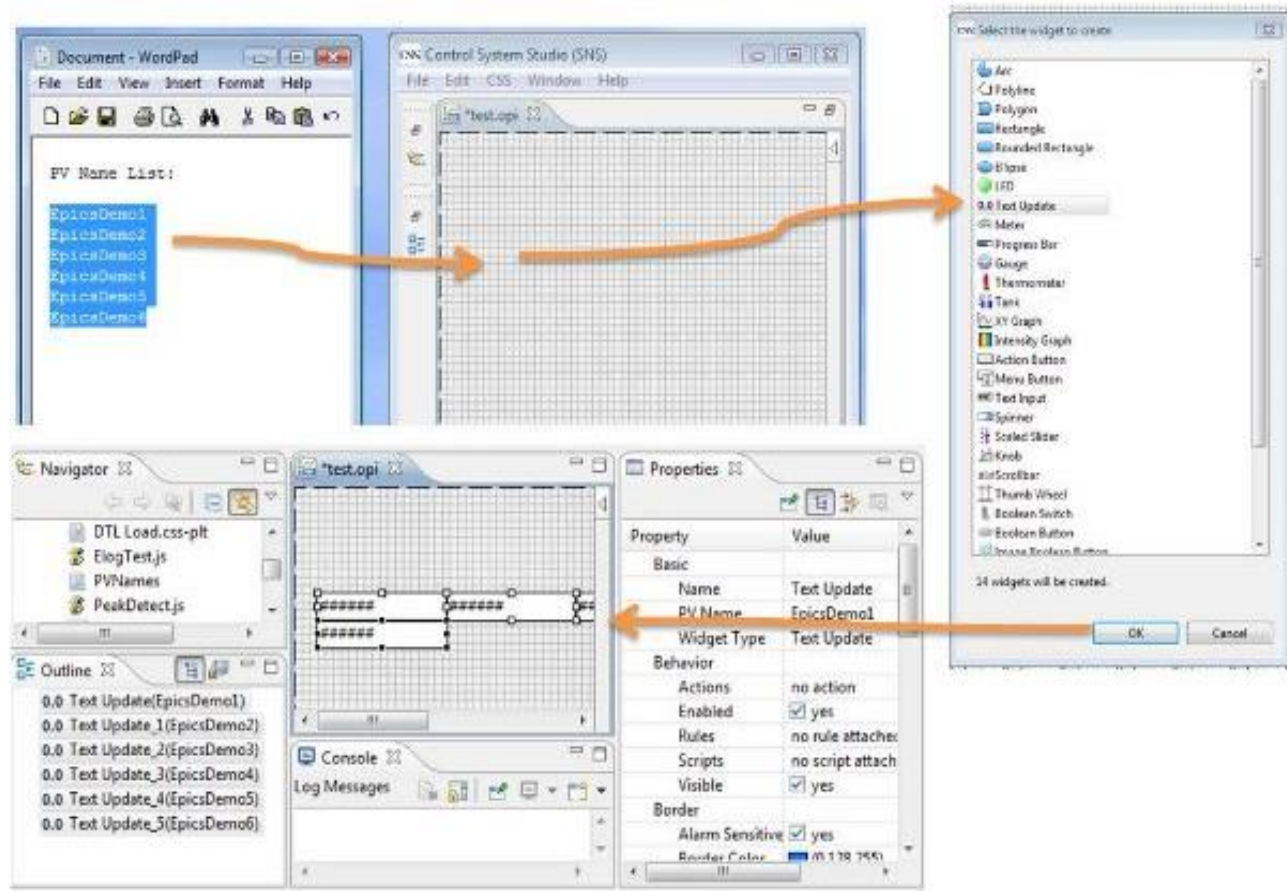
OPIs in ‘Tabs’

Try with OPI Examples: Open in tab, ... Window



# Hint: Drop PV Names

- Assume you have some text document with a list of PVs
- How to quickly create a display with Text Update widgets for these PVs?
  - Just drag the names into the display
  - Will be prompted for the type of widget



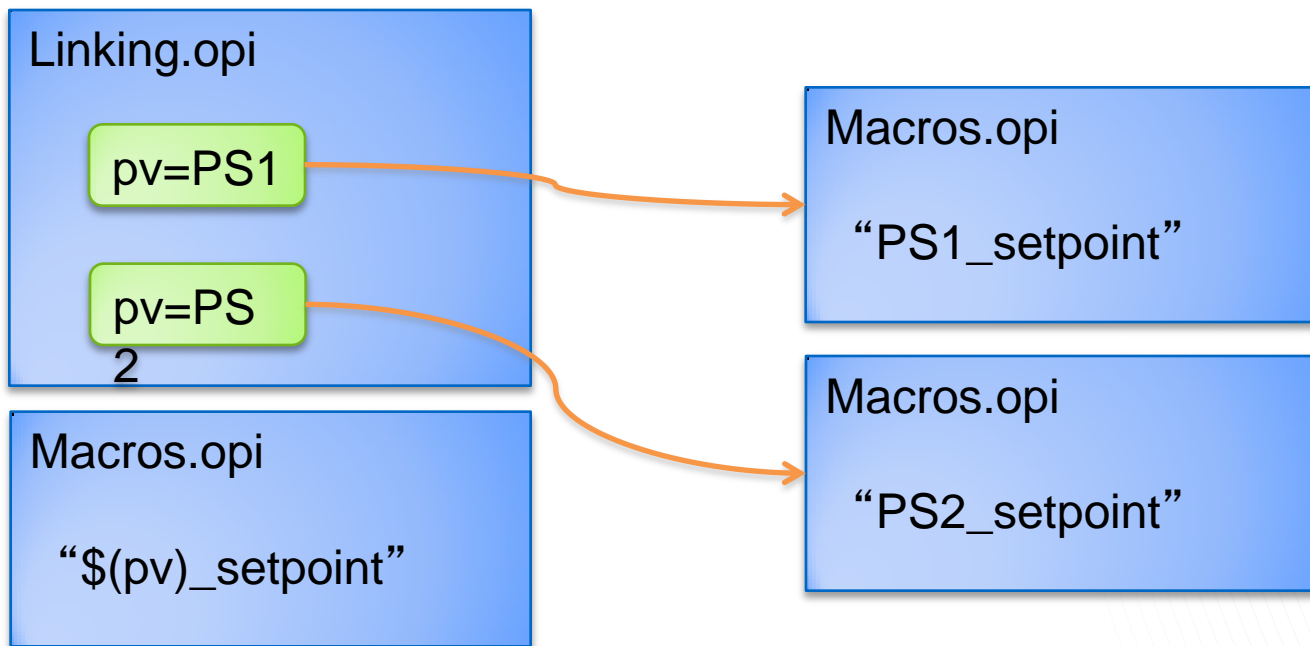
# Macros

Usage:  $\$(macro)$  or  $\${macro}$

- Wherever you enter a widget property
- Most often used for (partial) PV name:
  - $\$(pv)_setpoint$
  - $\$(pv)_readback$

Such a display can then be invoked with

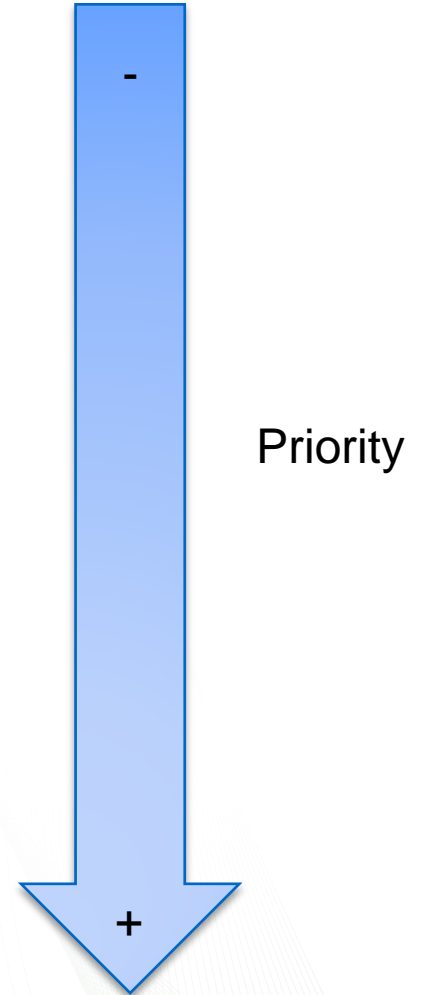
$pv=$ “PowerSupply1” or “PowerSupply2”



# Macro Definition

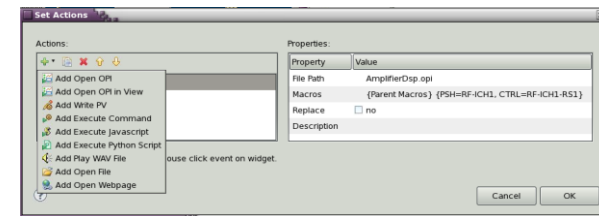
- Predefined Macros: Widget properties, see online help for name mapping
  - Property “X”: Macro \$(x)
  - Property “Name”: Macro \$(pv\_name)
  - Automatic: Macro \$(pv\_value)
    - See default for the “Tool Tip” property
- User-defined:
  1. BOY Runtime Preference Setting (-pluginCustomization .....
  2. User Preference settings (CSS, Pref..., ..App..., Display, BOY, OPI Runtime)
  3. Macro parameter of Action that opens the \*.opi file
  4. Display \*.opi file property “Macros”
  5. Grouping/Linking/Tabbed Container that wraps the widgets

Example:  
Macro parameter of *Action* will  
override *Preference* settings.



# Exercise: Linking Displays with Macros

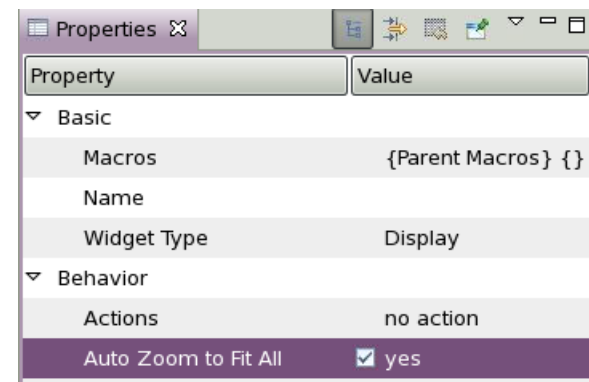
- Create display file “Macros.opi”
  - Label with Text “\$(pv)”
  - Text Update with PV Name “\$(pv)”
- Create display file “Linking.opi”
  - Action button with “Actions” to “Open OPI”
    - Use File Path for Macros.opi
    - Define Macros: pv= “sim://sine”
  - Add another action button (copy previous one)
    - Set macro to pv=“sim://ramp”
- Execute. Check that you can open the linked display
- Extra: Check OPI Examples, “4. Actions”
  - Can have more than one “Open OPI”
  - Any widget can have “Action”. Try Label.
  - Try Linking Container to display Macros.opi within Linking.opi





# Miscellaneous

- Display has an “Auto Zoom” property
  - Size will adjust to fit window

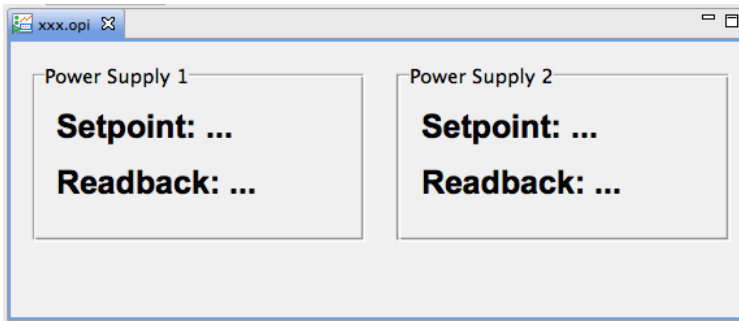


# Exercise: Grouping Container

In EDM, MEDM, ... we needed lines and rectangles to visually group related displays.

In BOY there is the Grouping Container

- Create a display with Grouping Containers that look like this:



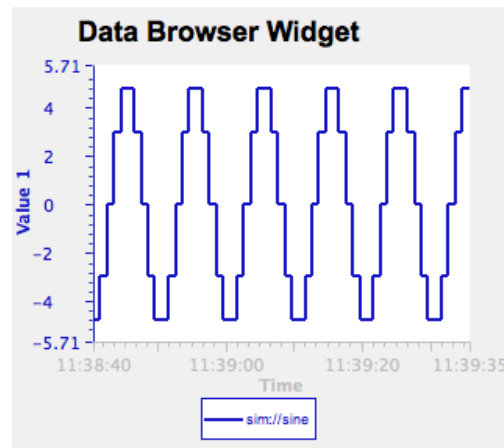
- Border Style=Group Box Style
  - Name = Power Supply 1, Power Supply 2
  - Add Labels “Setpoint:...”, “Readback:...”
- Note how you can
    - Move the Grouping Container and all its content
    - Move Labels inside and out of the container

# Exercise: “Striptool” type Plots of PV over Time

Try both options

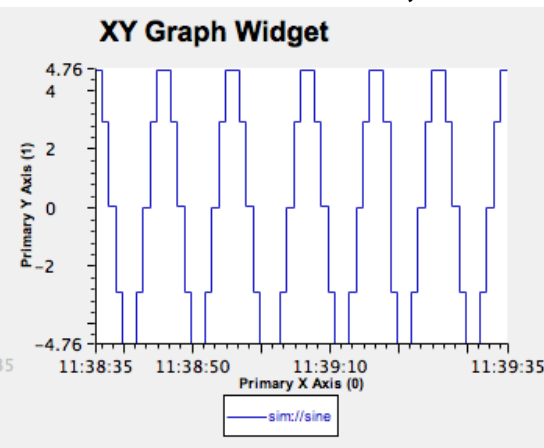
## – Data Browser Widget

- New Data Browser Plot, add PV
- Set desired axis and time range
- Save as \*.plt
- Add Data Browser Widget to BOY
- Set its File Name to the \*.plt



## – XYGraph Widget

- Behavior, Trigger PV: “sim://noise”
  - This PV updates once a second and will trigger plot updates
- Primary X Axis(0), Time Format: “HH:MM:ss”
  - To get a “time” axis
- Trace 0, Trace Type: Step Horizontally
- Trace 0, Update Mode: Trigger
- Trace 0, Y PV: Name of PV to plot



- ✓ Can also display archived data
- ✓ PV can be ‘monitored’, showing brief spikes

- ✓ Has many more display options
- ❑ Cannot show archived data
- ❑ PV scanned at update rate, can miss brief spikes

**TO BE  
CONTINUED...** 